An Efficient MapReduce Computing Model for Imprecise Applications

Changjian Wang, College of Computer, National University of Defense Technology, Changsha, China
Yuxing Peng, College of Computer, National University of Defense Technology, Changsha, China
Mingxing Tang, College of Computer, National University of Defense Technology, Changsha, China
Dongsheng Li, College of Computer, National University of Defense Technology, Changsha, China
Shanshan Li, College of Computer, National University of Defense Technology, Changsha, China
Pengfei You, College of Computer, National University of Defense Technology, Changsha, China

ABSTRACT

Optimizing the Map process is important for the improvement of the MapReduce performance. Many efforts have been devoted into the problem to design more efficient scheduling strategies. However, there exists a kind of MapReduce applications, named imprecise applications, where the imprecise results based on part of map tasks can satisfy the requirements of imprecise applications and thus the job processes can be completed when enough map tasks are processed. According to the feature of imprecise applications, the authors propose an improved MapReduce model, named MapCheckReduce, which can terminate the map process when the requirements of an imprecise application is satisfied. Compared to MapReduce, a Check mechanism and a set of extended programming interfaces are added to MapCheckReduce. The Check mechanism receives and analyzes messages submitted by completed map tasks and then determines whether to terminate the map phase according to the analysis results. The programming interfaces are used by the programmers to define the termination conditions of the map process. A data-prefetching mechanism is designed and implemented in MapCheckReduce which can improve the performance of MapCheckReduce effectively. The MapCheckReduce prototype has been implemented and experiment results verify the feasibility and effectiveness of MapCheckReduce.

KEYWORDS
Computing Model, Imprecise Application, MapCheckReduce, MapReduce

1. INTRODUCTION

MapReduce (Jeffrey and Sanjay 2004) is a kind of commonly used computing framework in cloud computing at present. MapReduce contains two main phases: map and reduce. The reduce tasks start processing data only when all the map tasks are completed because the input data of a reduce task may come from all the map tasks.

Improving the MapReduce performance is an important topic for researchers and many efforts have been devoted to it. Some works focus on task scheduling policies (Ali, Matei, et al. 2011) (Kay, Patrick, et al. 2013) (Aysan and Douglas 2011) (Joel, Deepak, et al. 2010) (Matei, Dhruva, et al. 2009) (Matei, Dhruva, et al. 2010). Some other works aim at the stragglers in MapReduce which are some slow tasks with long runtime significantly far behind most of the tasks of the same job.

In these research, it is still preserved that all the tasks should be processed in MapReduce. However, there exists a kind of special applications in MapReduce jobs which permit the imprecise results based on part of the input data. When enough map tasks are completed and the map outputs reach a certain size, it will bring little influence to the final result accuracy of these jobs to complete more map tasks. For example, the word frequency statistics and the hot-word detection for Internet public sentiment, both of them need to analyze vast numbers of text files. When enough map outputs have been generated, the statistical results will tend to be stable. On the other hand, the imprecise results based on part of map tasks are also able to meet the users’ requirements. These MapReduce applications can be named Imprecise Applications and we can improve the MapReduce performance in imprecise applications through terminating the map processes when enough map tasks are completed.

However, as mentioned-above, MapReduce is a batch-processing framework and no tasks should be given up during the MapReduce process. The common way is to choose a data subset and then to process it as a MapReduce job (Sameer Agarwal, et al. 2013) (Shivaram Venkatraman, et al. 2014) (Nikolay Laptev, et al. 2012). However, these approaches are to estimate the final results based on part of data and thus they may lead that the final results are not enough accurate and the requirements of users are missed.

In this paper, we propose an efficient computing model, named MapCheckReduce, to optimize the performance of MapReduce for imprecise applications. MapCheckReduce can significantly reduce the job runtime of imprecise applications.

MapCheckReduce reforms the map process in MapReduce. Besides the original data-processing on mappers, a new check process is added in. The check process receives and analyzes messages from mappers and then determines whether to terminate the map process and start the data-processing in the reduce tasks. The termination conditions of each imprecise application are also defined by users. In addition, MapCheckReduce implements the data-prefetching mechanism which can overlap the data-fetching phase and data-processing phase in adjacent tasks of the same mapper and thus hide the overhead to fetch data chunks from remote.

The contributions of this paper are as follows:

1. An improved MapReduce model, named MapCheckReduce, is proposed. The new model is designed to improve the performance of MapReduce for imprecise applications. MapCheckReduce can significantly reduce the job runtime of imprecise applications.

2. A new mechanism to terminate the map process, named Check, is proposed and added into the MapReduce framework. With the Check mechanism, the new MapCheckReduce framework can collect and analyze the messages from the completed map tasks and then determine whether to terminate the map phase.

3. A set of new programming interfaces are designed in MapCheckReduce, including check, sendMessage, acceptMessage and sendStop, and so on. With the interfaces, users can write their MapCheckReduce programs easily.
Related Content

On Measuring Cloud-Based Push Services
www.igi-global.com/article/on-measuring-cloud-based-push-services/144872?camid=4v1a

A Qualitative Approach to Understand Consumer Groups and Decision-Making Process in Online Group Buying: An Exploratory Study
www.igi-global.com/article/a-qualitative-approach-to-understand-consumer-groups-and-decision-making-process-in-online-group-buying/226288?camid=4v1a
Behaviour-Aware Discovery of Web Service Compositions
www.igi-global.com/article/behaviour-aware-discovery-web-service/3102?camid=4v1a

Learning Workflow Models from Event Logs Using Co-clustering
www.igi-global.com/article/learning-workflow-models-from-event-logs-using-co-clustering/100661?camid=4v1a